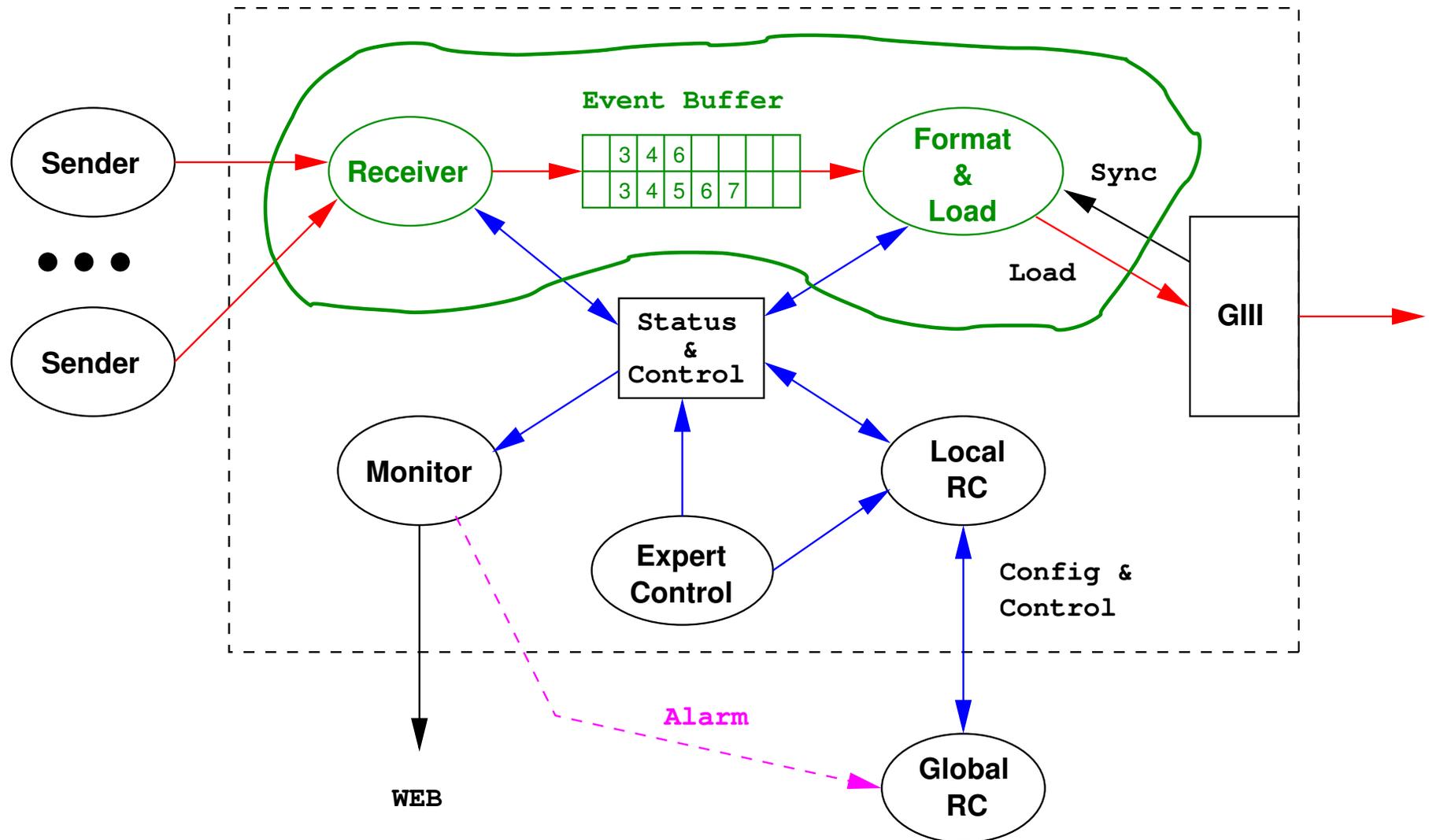


Frank Chlebana
Aug 28 2007

Adding Luminosity and Trigger Information to the Data Stream

Software Design/Requirements



Data Driven

Multiple “*Senders*” send event fragments (via xDAQ) to “*Receiver*” thread

Receiver moves event to a ring buffer

“*LoadG3*” removes event fragments from buffer, formats the data, and loads G3 card

“*Monitor*” thread collects statistics and writes status information to a shared memory area

Error conditions forwarded to central “*Error Handler*”

Toy Example

Toy example is written in “c”

Everything runs on one machine

Sockets used instead of xDAQ between Sender/Receiver

Server

→ Listens for connections on socket

Spawns multiple threads to accept events, one thread per client

→ Accepts event fragments and puts them into a buffer

→ Spawns thread to read ring buffer

→ Reads events from ring buffer

→ Spawns monitor thread, periodically updates shared memory

Client

→ Sends event fragments to server

Monitor

→ Reads shared memory and display statistics

Issues

→ Handle missing fragments, timeouts and out of order fragments

→ Rate is very slow...

Slow in sending event fragment... buffer is mostly free,
50 Hz, 200 KB/s

CD-ROM

Shell - Konsole <12>

Session Edit View Bookmarks Settings Help

```

monitor: time = 1187791908
monitor: events = 7891
monitor: rate = 206190.00 Bytes/s
monitor: rate = 49.00 Hz
monitor: fifoFree 0 71
monitor: fifoFree 1 100
monitor: fifoFree 2 100
monitor: fifoFree 3 100
monitor: fifoFree 4 100

```

Shell

Shell - Konsole <9>

Session Edit View Bookmarks Settings Help

```

[ch]lebanah@cmsroctest:example2]$ ipcs
----- Shared Memory Segments -----
key      shmid  owner      perms      bytes      nattch     status
0x000007d8 65537  ch]lebanah 666        60         2
----- Semaphore Arrays -----
key      semid  owner      perms      nsems
----- Message Queues -----
key      msqid  owner      perms      used-bytes   messages

```

Shell

Shell - Konsole <11>

Session Edit View Bookmarks Settings Help

```

client #0 sent nFrag 7700 size 2064
client #0 sent nFrag 7800 size 2064
client #0 sent nFrag 7900 size 2064
client #0 sent nFrag 8000 size 2064
client #0 sent nFrag 8100 size 2064
client #0 sent nFrag 8200 size 2064

```

Shell

Shell - Konsole <8>

Session Edit View Bookmarks Settings Help

```

client #1 sent nFrag 7800 size 2064
client #1 sent nFrag 7900 size 2064
client #1 sent nFrag 8000 size 2064
client #1 sent nFrag 8100 size 2064
client #1 sent nFrag 8200 size 2064

```

Shell

Shell - Konsole <4>

Session Edit View Bookmarks Settings Help

```

load3: nFrag = 7800/7798
load3: nFrag = 7900/7898
load3: nFrag = 8000/7998
load3: nFrag = 8100/8098
load3: nFrag = 8200/8198

```

Shell

Error Handling

Error conditions are recorded in the “Status and Control” block.

→ Define behavior for error conditions

→ Need to specify the recovery procedure

→ Specify the interaction with Run Control

Monitor process polls “Status and Control” block and communicates with central error handler...

- No connection

Sender cannot establish connection with receiver

- Lost connection

Sender loses connection with receiver in the middle of the run

- Missing event fragment

Event fragment from one sender is missing

- Fragment timeout

Event fragment from one sender does not arrive within timeout interval

- Fragment out of order

Event fragment from sender arrives out of sequence

- Buffer full

Event buffer is full

- G3 load timeout

Timeout while trying to load G3

Software Elements

Most of the software elements needed to develop the toy example into the real system are installed on cmsroctest

Need to install the software site-wide

TriDAS

Code to access G3 card

xDAQ

Used for transporting event fragment

RCMS

(mysql, tomcat)

RunControl and Monitoring System

Example to load G3

Transfers data between the two G3 cards in cmsroctest

```
[chlebana@cmsroctest ~/cms]$ ./merge
```

To run again :

```
/home/chlebana/TriDAS/daq/itools/packages/fedkit/test_merge -r 0 -s 0 -t 1  
-m 10 -M 10000 -n 10000 -c 0 -L 1 -o 0 -a 1 -T 0 -b 2048 -B 4096  
-h 0 -S 0 -R 0 -i 0 -F 1000 -g 0 -K 0 -J 0 -O 2
```

Successfully opened a FEDkit receiver with FPGA version 0x3f00002a

Successfully opened a FEDkit sender with FPGA version 0x3f200009

Received 10000 fragments. Still alive at Wed Aug 22 09:30:24 2007

...

* Summary * Total bytes = 398068008 in 0 s 772602 us

bw=515.230362MB/s

rate = 12.943275kHz

Received event number = 10000/10000

Avg event size : 39806.800800 bytes.

Average time per event = 77.260200 us

Rate = 500 MB/s

Data Format

Luminosity Data Format

`DataFormats/Luminosity/interface/LumiDetails.h`

`DataFormats/Luminosity/interface/LumiSummary.h`

Trigger Information

Not defined??

Other Issues

Mezzanine card

Can develop the system here... but will need to integrate into the rest of CMS... *Requires physical presence at CMS*

Long term support issues... pager carrier...

Summary

- Toy example demonstrates proof of principle
- Start replacing parts with more realistic software
 - *Use xDAQ for fragment transport*
 - *Transfer data to G3*
 - *Optimize performance/speed*
- Robust Error Handling and Recovery
- Finalize data format
- Add mezzanine card
- Interaction with RC
- Integration...